UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Adriano dos Reis Vieira

PROJETO DE DIPLOMAÇÃO

Otimização do protocolo EAPS - Ethernet Automatic Protection Switching , em topologias resilientes.

Porto Alegre
(2010)

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Otimização do protocolo EAPS - Ethernet Automatic Protection Switching , em topologias resilientes.

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para Graduação em Engenharia Elétrica.

ORIENTADOR: Carlos Eduardo Pereira

Porto Alegre

(2010)

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ADRIANO DOS REIS VIEIRA

EAPS

Este projeto foi julgado adequado para fazer jus aos créditos da Disciplina de "Projeto de Diplomação", do Departamento de Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof.Dr. Carlos Eduardo Pereira, UFRGS

Formação (Instituição onde obteve o título – Cidade, País)

Banca Examinadora:

Prof. Carlos Eduardo Pereira, UFRGS

Doutor em Engenharia Elétrica (Universidade de Stuttgart , Alemanha)

Prof. João Cesar Netto, UFRGS

Doutor em Ciências Aplicadas (Université Catholique de Louvain, Bélgica)

Engenheiro Luciano Lara, Global Crossing

Pós Graduado em Telecomunicações (Pontifícia Universidade Católica – PUCRS - Porto Alegre, Brasil)

Porto Alegre, (dezembro 2010).

DEDICATÓRIA

Dedico este trabalho aos meus pais, Nicanor e Noeli pelo apoio e motivação durante toda essa longa caminhada.

AGRADECIMENTOS

À Deus.

Ao professor Carlos Eduardo Pereira pela orientação e por suas preciosas horas de auxílio para conclusão deste trabalho.

Aos meus pais pelo carinho e motivação.

À minha amada esposa Vanessa Grbac, pelo carinho, dedicação e o apoio incondicional.

Aos colegas da empresa DATACOM pelo apoio durante o curso.

Aos colegas da Global Crossing, em especial ao Ary Aguiar, Luciano Lara e Emanuel Erichsen pela confiança no meu trabalho.

RESUMO

Este trabalho tem como objetivos melhorar o tempo de convergência e a confiabilidade do Ethernet Automatic Protection Switching (IETF RFC – 3619), melhorando a qualidade de serviço entregue ao assinante e melhorando a meta de indicadores de qualidade de uma operadora. Como objetivo principal, busca-se melhorar a resiliência minimizando a parada de tráfego. Através de análise de tempos de comutação utilizando um gerador de tráfego ethernet, são apresentadas as modificações necessárias no funcionamento do protocolo. São consideradas as características de topologias e os diferentes meios de interconexão do switches metro ethernet.

Palavras-chaves: EAPS. Ethernet. Resiliência. Protocolo. Switches. Metro Ethernet.

ABSTRACT

This report aims to improve convergence time and reliability of the Ethernet Automatic Protection Switching (IETF RFC - 3619), improving the quality of service delivered to the subscriber and the goal of improving quality indicators of an operator. The main objectives we seek to improve resilience by minimizing the traffic stop. Through analysis of switching times using an ethernet traffic generator, are presented the necessary changes in the functioning of the protocol. These are considered the characteristics of topologies and the different means of interconnection of Ethernet switches meters.

Keywords: EAPS. Ethernet. Resilience. Protocol. Switches. Metro Ethernet.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS

ARP: *Address Resolution Protocol*

BID: *Bridge ID*

BPDU: *Bridge Protocol Data Unit*

CRC: *Cyclic Redundancy Check*

CST: *Common Spanning Tree*

DA: *Destination Address*

DEC: *Digital Equipment Corporation*

FCS: *Frame Check Sequence*

CLI: *Command line interface*

EAPS: *Ethernet Automatic Protection Switching*

IETF: *Internet Engineering Task Force*

IEEE: *Institute of Eletrical and Eletronic Engineers*

DELET: *Departamento de Engenharia Elétrica*

MEF*: Metro Ethernet Fórum*

MSTP: *Multiple Spanning Tree Protocol*

NIC: *Network Interface Card*

STP: *Spanning Tree Protocol*

RSTP: *Rapid Spanning Tree Protocol*

RFC: *Request For Comments*

UFRGS: *Universidade Federal do Rio Grande do Sul*

VoIP: *Voice Over IP*

1 INTRODUÇÃO

A resiliência em redes Metro Ethernet tornou-se fundamental para garantir a qualidade de atendimento das operadoras aos seus assinantes. Em redes de computadores, resiliência é a capacidade de fornecer e manter um nível aceitável de serviço em caso de falhas e desafios de funcionamento normal[1]. O STP – *Spanning Tree Protocol* foi o primeiro protocolo a ser padronizado pelo IEEE em meados de 1990 [2], sendo aprimorado e chamado de RSTP quando foi padronizado em 1998 [3].

As exigências sobre os protocolos de resiliência tornaram-se ainda maiores com o passar dos anos. A maior capacidade dos backbones foi o fator que alavancou a pesquisa por protocolos mais rápidos e seguros. Em outubro de 2003 a fabricante de switches Extreme Networks, padronizou o EAPS – Ethernet Automatic Protection Switching, protocolo com maior rapidez de convergência e com a capacidade de proteção de múltiplas VLANs [5], neste mesmo ano foi padronizado o MSTP [4], cujo funcionamento assemelha-se ao RSTP, porém com a capacidade de proteger múltiplas VLANs.

O presente texto se encontra estruturado como segue: no Capítulo 2 é feita uma apresentação do funcionamento dos protocolos supracitados. O capítulo 3 apresenta detalhes de funcionamento do EAPS. A apresentação das melhorias realizadas no EAPS está demonstrada no capítulo 4.No capítulo 5 são apresentados os testes de desempenho e os resultados obtidos no protocolo EAPS. São mostrados os testes antes e após a melhoria do código. O trabalho encerra-se no capítulo 6 com as conclusões gerais do estudo realizado.

2 PROTOCOLOS DE RESILIÊNCIA

2.1 STP – Spanning-Tree Protocol

Especificado pela norma da IEEE 802.1D o propósito do STP - protocolo Spanning-Tree é criar dinamicamente uma rede com bridges e switches em que exista apenas um caminho ativo entre um par qualquer de segmentos de rede (Domínios de Colisão) [5]. Para atingir este objetivo, todas as bridges e switches usam um protocolo dinâmico. O resultado deste protocolo é que cada interface de uma bridge irá ficar em um estado de "blocking"ou de "forwarding". "Blocking" significa que uma interface não pode enviar ou receber frames, mas ela pode enviar e receber CBPDUs – Configuration Bridge Data Units. "Forwarding" significa que o dispositivo pode enviar e receber frames. Colocando o conjunto correto de portas em estado "Blocking" é possível criar um único caminho lógico entre um par de redes.

Como sabemos se múltiplas conexões entre switches são criadas para redundância, "loops" na rede podem ocorrer, aumentando o congestionamento na rede, o STP (*Spanning-Tree Protocol*) foi criado com o intuito de parar os "loops" e permitir a redundância.

Os principais benefícios do Spanning-Tree são:

- É possível ter links fisicamente redundantes, que podem ser usados quando outro link falhar - resiliência;
- A lógica da bridge é confundida com múltiplos caminhos ativos para o mesmo endereço MAC, o STP evita isto criando um único caminho;
- Loops em uma bridge são evitados.

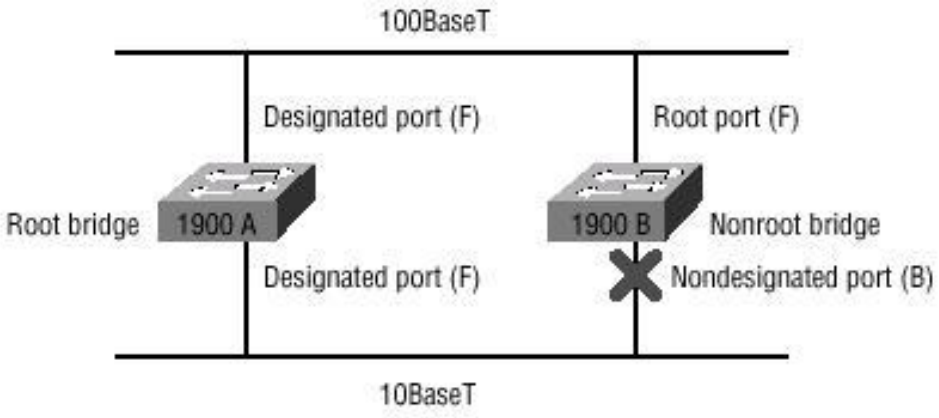


Figura 2.1 Eleição da porta raiz

O STP encontra todas as conexões na rede e derruba todas as conexões redundantes, com isso qualquer "loop" que poderia estar ocorrendo na rede é eliminado. O STP resulta em cada uma das portas sendo colocada em um de dois estados "forwarding" ou "blocking". A forma como ele faz isso, é elegendo uma "ponte raiz" (*root bridge*) que irá decidir sobre a topologia de rede, pode-se ter somente uma *root bridge* em uma rede. As portas desta *root bridge* são denominadas "portas designadas" (*designated ports*), que estão em estado de operação chamado de "modo de encaminhamento" (*forwarding-state*), que enviam e recebem o tráfego da rede.

Outros switches na rede são chamados de "pontes não-raiz" (*nonroot-bridge*), entretanto a porta com menor custo para a *root bridge* são chamadas de "porta raiz" (*root port*), estas portas também enviam e recebem o tráfego na rede.

As portas com "menor custo de caminho" (*lowest-cost path*) para a *root bridge* são as *designated ports*, as outras portas são chamadas de "portas não designadas" e estão em estado de operação chamado "modo de bloqueio" (*blocking state*), neste modo estas portas não enviam e não recebem o tráfego da rede. Switches e bridges que rodam o protocolo STP

trocam informações que são chamados BPUD (*Bridge Protocol Units Data*). BPUDs enviam mensagem com configuração utilizando frames multicast. O ID de cada dispositivo é enviado para os outros dispositivos através das BPUDs, a cada 2 segundos, este ID é utilizado para determinar quem será a *root bridge*, pois neles temos dois campos importantes, prioridade e o endereço MAC do dispositivo. A prioridade default em todos os dispositivos rodando o protocolo STP IEEE é 32768 (0x8000) [2].

Para determinar a *root bridge* é feita uma combinação dos campos endereço MAC e prioridade. Se dois switches tem a mesma prioridade o switch com o menor endereço MAC será a *root bridge*. Por exemplo, temos um switch com prioridade 0x8000 e endereço MAC:0000.0C00.1111.1111 e outro switch com mesma prioridade e endereço MAC:0000.0C00.2222.2222, neste caso o primeiro switch será a *root bridge*.

Será observado no CLI do equipamento que o campo "*Cost of Path to Root*" está com valor zero, isto indica que esta BPUD é de um switch que atualmente é a *root bridge*. A tabela 2.1 a seguir mostra os custos das portas conforme definido pela IEEE originalmente e atualmente.

Velocidade	Nova IEEE custo	Original IEEE custo
10Gbps	2	1
1Gbps	4	1
100Mbps	19	10
10Mbps	100	100

Tabela 2.1 Custos das interfaces

A seguir serão demonstrados os parâmetros contidos nos BPDUs trocados entre os sw*itches*. A Figura 2.2 ilustra o formato do BPDU.

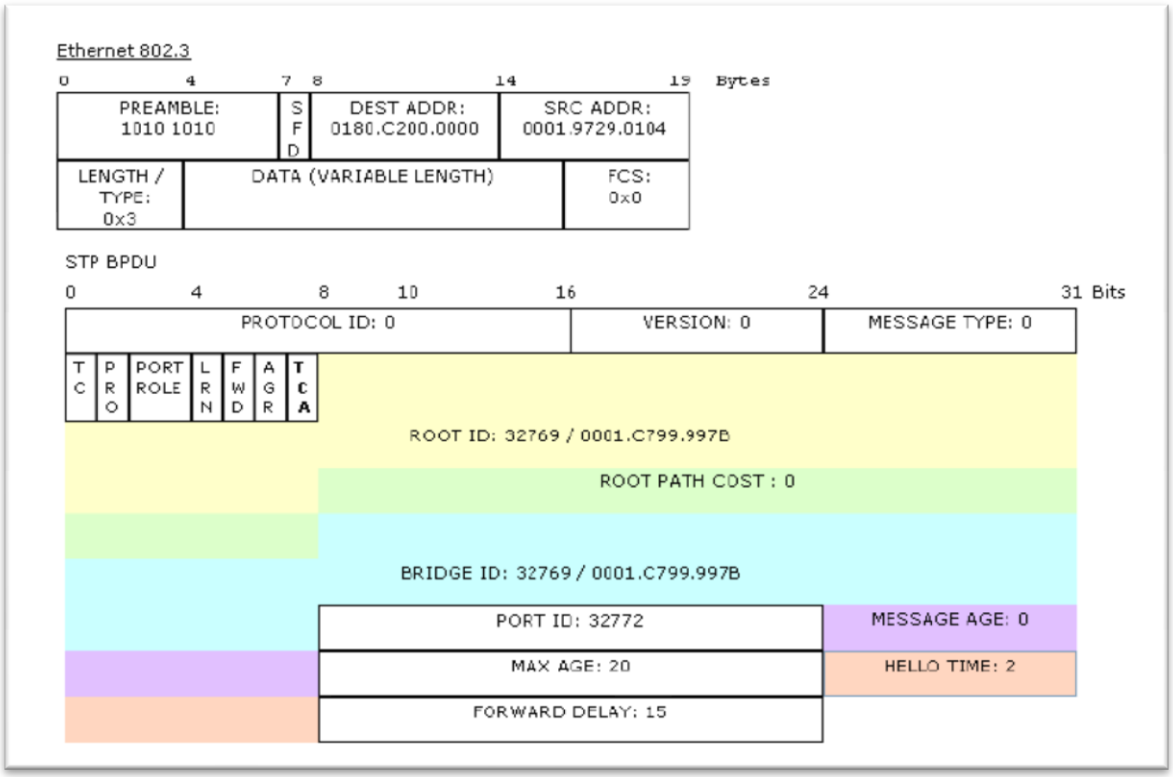


Figura 2.2 Formato da BPDU de configuração [ref]

A figura 2.2 foi capturada do software *Packet Tracer* para facilitar a compreensão dos parâmetros do *Spanning Tree*. Observa-se que se trata de um cabeçalho Ethernet 802.3 e, seus respectivos campos. O endereço de origem corresponde ao MAC da porta individual para cada BPDU, já que para cada porta do switch há diferentes endereços físicos. Portanto, esse endereço diferencia-se do usado para criar o *Bridge* ID, que se trata de um endereço global. Cada campo do cabeçalho está descrito abaixo:

• *Protocol ID:* Identifica o algoritmo *Spanning Tree* e o protocolo. Consiste de 2 *bytes*;

• *Version*: Identifica a versão do protocolo e consiste de apenas 1 *byte;*

• *Message Type:* Determina qual dos dois formatos BPDUS esse *frame* contém; configuração e notificação de alteração da topologia. Seu tamanho equivale a 1 *byte*;

• *Flags:* Indica os índices de BPDU em caso de uma mudança da topologia;

As Flags existentes são: TC (*Topology Change Nofitication*), que é usado pelo switch raiz para reconhecimento de mudança de topologia e TCA (*Topology Change Notification Acknowledgment*), o qual diz ao switch que os dados da topologia contidos no BPDU atual foram lidos e salvos;

• *Root ID:* Contém o *Bridge* ID do switch raiz. Depois que ocorre a convergência, todos os BPDU configuram esse campo para identificar o switch raiz. Consiste de *8 bytes*;

• *Root Path Cost:* Indica o custo acumulado para o switch raiz, a fim de detectar a melhor opção para transmitir BPDU ao switch principal. Consiste de 4 *bytes*;

• *Bridge ID:* É o identificador gerado pelo BPDU para o switch e é usado pelo algoritmo para construir o *Spanning Tree*. Consiste de 4 *bytes*;

• *Port Id:* Contém um único valor para cada porta, assim a porta 1/1 possui o valor 0x8001 e a porta 1/2 contém 0x8002. É utilizado para identificar a porta do switch que emite mensagens para outros. Consiste de 2 *bytes*;

• *Max Age:* Designa a idade do BPDU que é acumulado com o tempo decorrido desde que o switch originou-o. Se há perda de conectividade com o switch principal e, portanto, para de receber BPDU, esse tempo é incrementado para identificar que esses dados são velhos. Ou seja, é o tempo que o switch espera antes de concluir que a topologia modificou. Consiste de 2 b*ytes*;

• *Forward Delay:* Tempo em que a porta fica em um determinado estado. Consiste de 2 *bytes*;

• *Message Age:* Intervalo de tempo em que o switch anuncia o BPDU. Consiste de 2 *bytes*;

• *Hello Time:* Tempo gasto em que o switch publica o BPDU, correspondente a 2 segundos.

O TCN BPDU é a notificação de alteração da topologia mais simples do que o BPDU de configuração e, como seu nome sugere, sua principal função é notificar sobre a alteração da topologia. A diferença encontra-se no tamanho do cabeçalho do TCN BPDU, que é identificado pelos três primeiros campos do BPDU de configuração: o *protocol* ID, *Version* e o *Type*. Esse último contém dois valores:

• 0x00: equivale a 0000 0000 em binário, que identifica o BPDU de Configuração;

• 0x80: equivale a 1 000 0000 em binário utilizado para identificar o TCN BPDU.

2.2 RSTP – Rapid Spanning-Tree Protocol

O funcionamento do RSTP é semelhante ao STP [4].

As principais características de aprimoramento do RSTP são [4] [5]:

• Organiza os segmentos da rede em árvore num tempo de na ordem de dezenas de milissegundos.

• Suporta fast-forwarding nas extremidades da rede – a porta da borda da rede pode ser configurada como edge-port, dessa maneira não irá aguardar por BPDUs de um possível NIC.

• Suporta switches com mais de 256 portas.

• Compatível com STP.

• Os BPDUs contêm informações adicionais, como mostra a figura 2.3 a seguir.

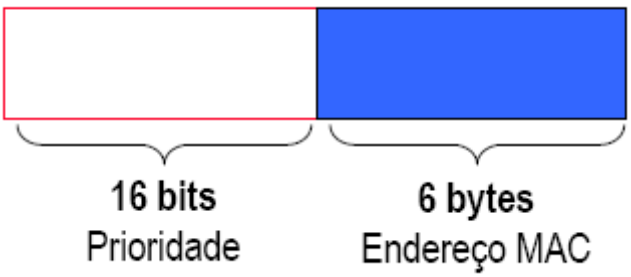


Figura 2.3 BPDU do RSTP

• Aproveita o fato de as ligações serem ponto a ponto.

• O RSTP determina que as Portas Alternativas (PA) irão receber os BPDUs que não são tão “bons” quanto os recebidos pela porta raiz. Assim, se um switch para de receber BPDUs do switch raiz, RSTP escolhe a melhor PA como nova porta raiz, acelerando assim o processo de convergência [5].

• A outra porta nova (BP) é usada somente quando um único switch tem dois enlaces para o mesmo segmento. Para que isso aconteça o switch deve estar conectado a um hub, como mostra a Figura 2.34. O princípio de funcionamento é o mesmo: O switch coloca uma das portas no estado FW e a outra no DC. Com isso o switch envia BPDUs pela porta FW e recebe os mesmos na porta DC. Então ele sabe que tem uma conexão extra para aquele segmento, assim chamada de Backup Port (BP). Caso a porta que está no estado FW falhe, o RSTP no switch coloca imediatamente a BP no estado FW.

• A possibilidade de colocar uma porta que anteriormente estava bloqueada (BP ou PA) imediatamente no estado FW, sem ter que passar pelos estados intermediários Listening (LN) e Learning (LR), é a razão do menor tempo levado pelo RSTP para convergir [5]. A Tabela 2.2 faz um comparativo dos estados do STP versus RSTP, demonstrando o melhor desempenho do RSTP entre os estados quando habilitado. Havendo estados intermediários para estados em que o protocolo está operacionalmente habilitado ou desabilitado, estes fazem com que diminua o desempenho de qualquer protocolo de resiliência [6].

Estado Operacional	Estado STP	Estado RSTP
Disable	Disable	Discarding
Enable	Blocking	Discarding
Enable	Listening	Discarding
Enable	Learning	Learning
Enable	Forwarding	Forwarding

Tabela 2.2 Comparativo do estado das portas.

2.2.1 Alteração de Topologia: STP versus RSTP

• **Alteração de topologia em STP**

- Qualquer alteração de estado de uma porta gera um TCN

- Quando uma bridge detecta alteração de topologia (TC) envia um TCN (topology change notify) para a root bridge

- A Root Bridge envia TC (BPDUs de configuração)

- Quando uma Bridge recebe estes BPDU diminuiu o aging time para o forward delay (para refrescar a tabela de endereços MAC)

• **Alteração de topologia em RSTP**

- Apenas as portas non-edge passarem para forwarding original TC BPDU.

- A falta de conectividade não se considera mudança de topologia. Neste caso uma porta passar para o estado Blocking não gera TC BPDU.

- Os switches limpam a tabela MAC e passam ao estado Forward quase imediatamente.

- A propagação de mudança de topologia é processo de um passo.

- Quem inicia a mudança de topologia envia continuamente a informação em vez de ser apenas a root para fazer.

- O RSTP age de maneira pró ativa não existem os timers de delay existentes no STP.

- O algoritmo de eleição do Root Brigde no RSTP ocorre da mesma maneira que o STP.

2.3 MSTP – Multiple Spanning Tree Protocol

O Multiple Spanning Tree protocol (MSTP) carrega o conceito da IEEE 802.1w – Rapid Spanning Tree Protocol (RSTP) com o avanço de permitir ao grupo associar. VLANs a várias instâncias de spanning tree [5]. O protocolo utiliza uma VLAN para controle, a adição disto proporciona uma convergência mais rápida e ainda a possibilidade de balanceamento de carga. Cada MSTI - Multiple Spanning Tree Instance pode ter sua própria topologia independente. A multiplicidade de caminhos de encaminhamento fornecido por esta arquitetura melhora a tolerância a falha de rede pois, se uma instância falhar o fluxo de dados continua inalterado ao longo dos demais caminhos. Pode-se gerenciar grandes redes e usar caminhos redundantes mais facilmente.

Switches que rodam MST - Multiple Spanning Tree oferecerem interoperabilidade com um SST - Single Spanning Tree da seguinte forma [6] :

• MST- Multiple Spanning Tree executam IST - Internal Spanning Tree. O IST adiciona informações internas sobre a região do MST para o CST – Commom Spanning Tree.

• O IST conecta todas as portas ponte na região do MST e surge como uma sub-árvore na CST, que inclui todo o domínio de switches[9].

• O CIST - Commom Internal Spanning Tree é o conjunto das seguintes características:

- TSI - Internal Spanning Trees em cada região do MST;

- O CST - Common Spanning Tree, que interliga as regiões do MST;

- As pontes de SST – Single Spanning Tree;

- Dentro de uma região de MST a CIST é idêntica a uma IST. Fora de uma região MST a CIST é idêntica a uma CST.

- O STP, RSTP e MSTP juntos elegem uma única porta como a *root* (raiz) da CIST.

Dentro de cada região do MST são estabelecidas e mantidas as instâncias de MST (MSTIs). Estes são adicionalmente calculados por árvores geradoras de MSTP para fornecer uma forma simples e totalmente ativa de conexões. numeradas de 0, e o MSTIs são numeradas 1, 2, 3, e assim por diante.

Termo	Acrônimo	Definição
Boundary Port	-	Uma porta ponte anexada a uma ponte MST de uma LAN, que está em outra região.
Common Spanning Tree	CST	O único Spanning Tree calculado pela STP, RSTP, e por MSTP para conectar as regiões do MST. As regiões MSTP aparecerm como um ponte virtual simples na CST.
Common and Internal Spanning Tree	CIST	Uma coleção de ISTs em cada região do MST , e a árvore comum abrangendo (CST), que interliga as regiões do MST. A CIST é calculada pela MSTP para garantir que todas as redes locais na ponte local da rede são simples e totalmente conectadas.
Internal Spanning Tree	IST	A conectividade oferecida pela CIST dentro de uma dada Região MST. O IST é o MSTI em primeiro lugar na região,

		numerada como MSTI0, e existe por padrão e não pode ser removido. Todas as outras instâncias do MST são numeradas de 1 a 15
Multiple Spanning Tree Instance	MSTI	Uma de um número de Spanning Trees calculadas por MSTP dentro de uma Região do MST. O MSTI é definido por cada VLAN grupo, e é projetado para proporcionar uma forma simples e totalmente conectada a ativos da topologia de quadros classificados como pertencentes a VLANs são mapeados para o MSTI pela MST tabela de configuração que é usado pelo MST Pontes daquela Região MST.
MST Configuration Table	-	Uma tabela configurável que aloca cada VLAN para o Spanning comum
MST Bridge	-	Uma ponte capaz de suportar a CST e um MSTIs ou mais, e seletivamente mapear quadros classificados em qualquer VLAN dado à CST ou um determinado MSTI.
MST Configuration Identifier	MCID	Um nome, nível de revisão, e um sumário de uma dada a atribuição de VLANs de Spanning Trees.

MST Region	-	Um conjunto de LANs e MSTs bridges fisicamente ligadas através de portas em pontes, onde cada LAN CIST designada a uma ponte de MST, e cada porta é tanto a porta designado em uma das LANs, ou então uma não-Designado porta de uma ponte MST que é ligado a uma das LANs, cuja MCID corresponda exatamente ao MCID da porta Designada da LAN.
Single Spanning Tree Bridge	SST Bridge	Uma ponte capaz de suportar apenas um único spanning tree, a CST. Uma única árvore spanning pode ser suportado pelo Algoritmo Spanning Tree Protocol (STP) ou pelo Algoritmo de Rapid Spanning Tree Protocol (RTSP).

Tabela 2.3 Definições e acrônimos.

2.3.1 Múltiplas Regiões de Spanning tree

Para configurar switches que irão participar de várias instâncias de spanning-tree (MSTIs), deverá ser consistentemente configurado as opções com as mesmas informações de configuração do MST. Os switches interconectados que têm a mesma configuração do MST

compreende uma região MST [0], exceto para o caso de os switches serem ligados através de uma mídia compartilhada (ou seja, LAN).

A configuração do MST determina para qual região MST cada chave pertence. A configuração inclui o nome da região, o número de revisão, e a instância MST . A região pode ter um membro ou vários membros com a mesma configuração do MST. Cada membro deve ser capaz de processar BPDUs RSTP. Não há limite para o número de regiões do MST em uma rede, mas cada região pode suportar até 16 instâncias spanning-tree. É possível atribuir uma VLAN a apenas uma instância de spanning-tree por vez.

2.3.2 IST e CIST

IST é o Spanning tree interno, que roda na região MST**.** Dentro de cada região do MST, o MSTP mantém várias instâncias spanning-tree. A instância 0 é um caso especial de uma região, conhecida como a árvore interna abrangendo o IST. Todas as outroa instâncias do MST são numerados de 1 a 15. O IST é a única instância spanning-tree que envia e recebe BPDUs. Todas as informações de outras instâncias spanning-tree estão contidas em M-registros, que são encapsulados dentro MSTP BPDUs, pois o BPDU MSTP carrega a informação para todas as instâncias, o Multiple Spanning Tree Protocol o número de BPDUs que precisam ser processados por um switche para suportar múltiplos casos de spanning-tree é significativamente reduzida. Todas as instâncias do MST na região partilham os mesmos temporizadores mesmo protocolo, mas cada instância MST tem seus próprios parâmetros de topologia, tais como root switch ID, custo do caminho da raiz, e assim por diante. Por padrão, todas as VLANs são atribuídas ao IST. Um exemplo do MST é o local para a região, por

exemplo, uma instância do MST na região A é independente de uma instância do MST na região B, mesmo que as regiões A e B estão interligadas.

CIST é um spanning-tree comum e interno abrangendo uma coleção de TSI em cada região MST, é o spanning tree comum abrangendo (CST), que interliga as regiões MST para um único spaning tree gerador. O spaning tree gerador calculado em uma região aparece como uma sub da CST, que abrange todo o domínio comutado. A CIST é formada como o resultado de algoritmo de árvore geradora de execução entre os switches que suportam o 802.1W, 802.1s, 802.1D e protocolos. A CIST dentro de uma região do MST é a mesma que a CST fora de um região.

2.3.3 Operações dentro de uma Região do MST

O IST conecta todos os switches na região do MSTP. Quando o IST converge, o root da IST se torna o mestre do IST , que é o switche dentro da região com o mais baixo ID da bridge e custo do caminho para o root da CST. O mestre IST é também o root da CST se houver apenas uma região dentro da rede. Se o root da CST está fora da região, um dos MSTP muda na fronteira da região é selecionado como o IST mestre.

Quando um detector MSTP inicializa, ele envia BPDUs afirmando-se como o root da CST e o comandante do IST, com ambos os custos do caminho para o root da CST e ao mestre IST definido para zero.

O switch também inicializa todas as suas instâncias do MST, e afirma ser root de todos eles. Caso o switch receba informações do root superior MST (ID ponte mais baixa, menor custo de caminho) do que atualmente armazenados para a porta, ele abandona a sua eleição como o mestre IST.

Durante a inicialização, uma região pode haver muitas sub-regiões, cada uma com seu

próprio mestre IST. Como switches recebem informações IST superior, eles deixam suas sub-regiões anigas e aderem a nova sub-região que possam conter o mestre IST verdadeiro. Assim, todas as sub-regiões , exceto para o comandante que é aquele que contém o mestre IST verdadeiro. Para um funcionamento correto, todas as chaves na região MST devem concordar com o mesmo mestre IST. Portanto, todos os dois switches na região irão sincronizar suas funções de porta para uma instância MST somente se eles convergem para um mestre IST comum.

Somente a instância CST envia e recebe BPDUs e a instâncias do MST adiciona a seu spanning-tree informações para o BPDUs para interagir com switches vizinhos e calcular a rota final abrangendo a topologia do spanning tree. Devido a isso, os parâmetros de spanning-tree relacionadas com BPDU de ransmissão (por exemplo, hello, aging, max-age, e-max) são configurado somente na instância CST, mas afetam todas as instâncias do MST. Parâmetros relacionados com a topologia spanning-tree (por exemplo, a prioridade da chave, o custo da porta, a prioridade da porta) podem ser configurados tanto para a instância CST como para a instância MST.

O IST e instâncias do MST não usam a mensagem de tempo e informações sobre o aging máximo no BPDU de configuração para calcular a topologia de árvore estendida. Em vez disso, eles usam o custo do caminho para root e um mecanismo de contagem de saltos, semelhante ao mecanismo IP time-to-live (TTL).

Ao utilizar o comando *máximo MSTP-hops*, no modo de configuração, é possível configurar o máximo de saltos dentro da região e aplicá-lo ao IST e todas as instâncias na região MST. A contagem de saltos alcança o mesmo resultado que a mensagem (determina quando disparar uma reconfiguração). A chave de raiz da instância sempre envia uma BPDU (ou M-record) com um custo de 0 e a contagem de saltos definido para o valor máximo. Quando um switch recebe essa BPDU, ele diminui a contagem de saltos restantes recebido

por um e propaga este valor como a contagem de saltos restantes no BPDUs que gerado. Quando a contagem atinge zero, o switch descarta o BPDU aging e as informações mantidas para a porta.

2.3.4 MSTP Fast Ring Rapid Convergence

O MSTP Fast é uma opção para rápida convergência em anel, que impede as portas uplink do anel de aprendizagem de qualquer novo endereço em suas portas de uplink. Esta opção é altamente recomendável para maximizar o desempenho e a estabilidade da rede.

2.3.5 Spanning Tree IGMP Fast Recovery

Usando a recuperação rápida STP IGMP, o tráfego Multicast aproveita a conectividade de tempo de convergência fornecidos peloprotocolo Spanning Tree . Na Figura2.4 todos os switches executam o IGMP snooping e qualquer um dos Spanning Tree Protocolos - RSTP STP, MSTP [6]. O router faz flood do tráfego multicast na topologia de grupos multicast, para que um novo cliente seja inserido na topologia. A assinatura de um grupo multicast específico é feita através do envio de relatórios IGMP.

O roteador multicast envia uma consulta IGMP para os clientes para o seu grupo multicast associações. IP hosts responder com IGMP Reports. O tráfego flui a partir do roteador, através de Switch Switch D e A, para mudar C.

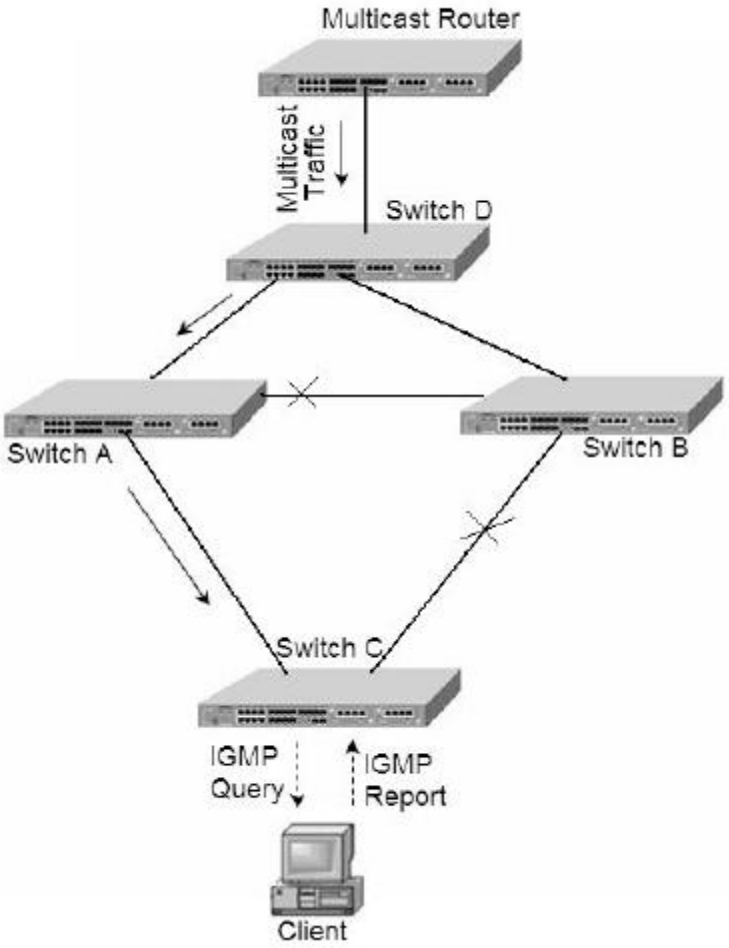


Figura 2.4 Spanning tree Fast Recovery.

O roteador multicast envia uma consulta IGMP para os clientes e para o seu grupo multicast . Hosts IP respondem com IGMP Reports. O tráfego flui a partir do roteador, através de Switch Switch D e A, para mudar C.

A porta mrouter (uma porta mrouter é uma porta que recebeu uma consulta IGMP de uma dispositivo upstream) do Switch C, que liga a chave B é bloqueado. Se ocorre uma mudança de topologia ea ligação entre Switch C e A a chave desce, e a porta bloqueada do Switch C volta ao seu estado de encaminhamento. Se o Spanning Tree IGMP fast recovery é configurada no switch C, o switch reage à mudança de topologia, enviando um IGMP de

consulta geral a todos os seus mrouters.

O cliente para de responder a consulta geral com um report de IGMP. O switch C encaminha o report IGMP a suas portas e o merouter report vai para o roteador multicast através dos Switches B e D. Como resultado, o tráfego do cliente ligado ao switch C é transmitida através do Switch B através do Switch A, como mostrado na figura 2.4.

3 EAPS - ETHERNET AUTOMATIC PROTECTION SWITCHING

Em respeito à política de privacidade e da concordância no que diz respeito à propriedade intelectual da empresa, onde foi realizado este trabalho de aprimoramento, não serão apresentados trechos de códigos tão poucos detalhes do hardware ou software, produzido pela empresa. Assim como o conteúdo desse trabalho será utilizado de forma puramente acadêmica, não podendo ser publicado ou reproduzido por nenhum meio. Serão respeitados todos os possíveis conflitos de interesses que possam surgir após a apresentação deste, como ganhos pessoais ou qualquer mérito de cunho pessoal. As cópias fornecidas à banca não poderão ser doadas para bibliotecas, outros estudantes ou ainda para terceiros. A empresa entende que o fato de um concorrente ter apenas as instruções contidas neste trabalho já seria suficiente para que fosse tentado chegar a este mesmo resultado. A melhoria do tempo de convergência do EAPS representa um valor agregado considerável no produto final, por este motivo a empresa solicita o cuidado com o descarte do material apresentado.

A indústria de telecomunicações tem contado com o Protocolo Spanning Tree (STP) em grandes redes de Camada 2 para fornecer um certo nível de redundância. No entanto, os xSTPs são insuficientes para fornecer os níveis de resiliências necessárias em tempo real para aplicações de uso crítico. É importante notar que toda a indústria reconheceu que uma nova tecnologia é necessária para substituir o xSTP e muitos vendedores estão no processo de desenvolvimento de tecnologias para atender a essa exigência.

O EAPS é uma solução desenvolvida pela Extreme Networks para tolerância a falhas L2 em topologias em anel. O EAPS garante redes livre de loops e com baixo tempo de recuperação, sendo ideal para uso em aplicações que exijam alta disponibilidade.

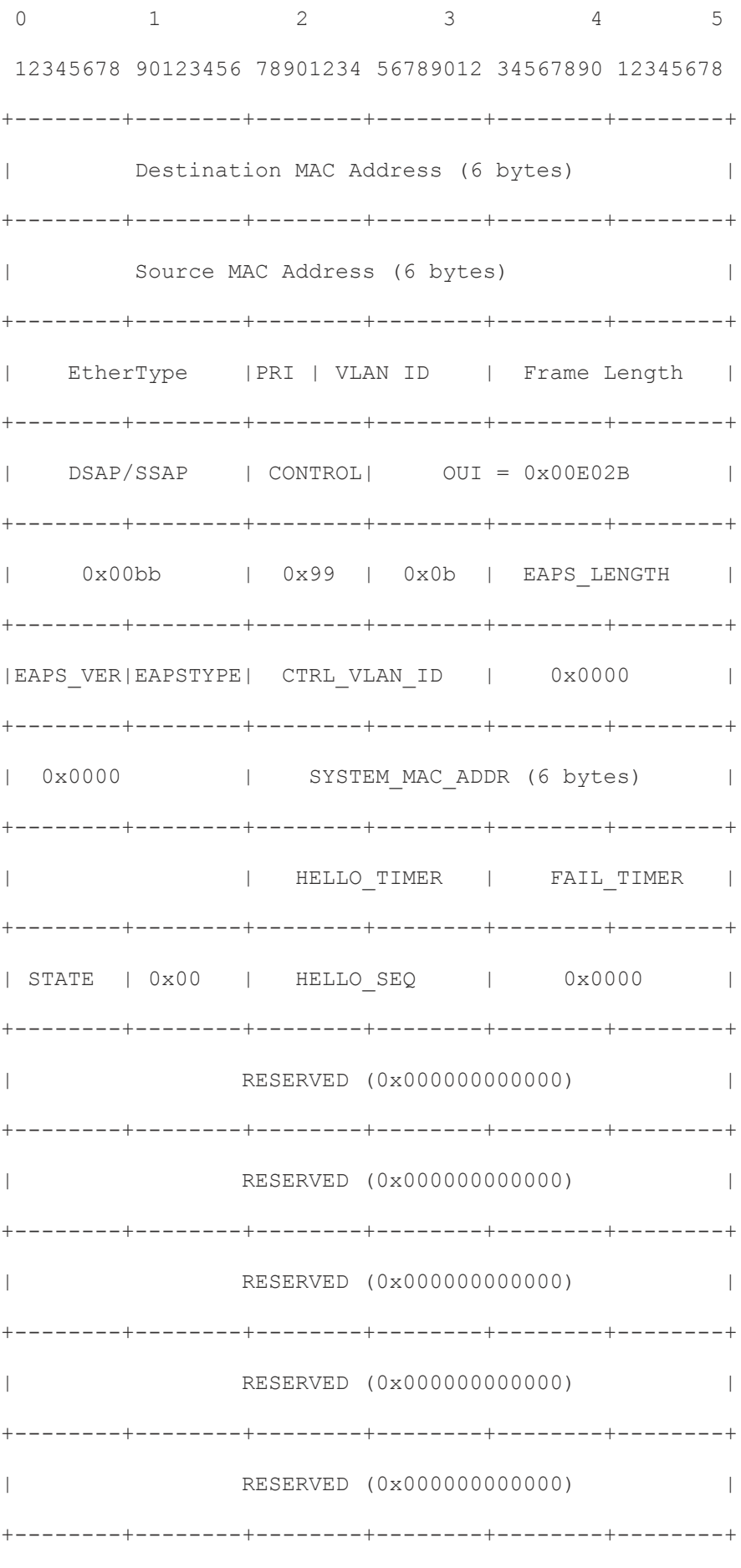
3.1 Definição

O anel é composto por dois ou mais switches, sendo um deles denominado Master e os outros Transit. Cada switch tem duas portas que pertencem ao anel, a primária e a secundária. Um domínio EAPS protege um grupo de VLANs, chamadas protected VLANs ou VLANs protegidas. Poderá haver múltiplos domínios EAPS rodando num mesmo switch e é obrigatório que cada domínio tenha uma VLAN de controle exclusiva para troca de mensagens EAPS.

3.2 Operação

Para que o EAPS entre em funcionamento é necessário criar ao menos um domínio. A esse domínio devem ser adicionadas as protected VLANs, a VLAN de controle e as portas primárias e secundárias, que devem pertencer as protected VLANs. Essa configuração deve ser feita em todos os switches do anel e apenas um deles deve ser configurado como Master.

O MAC de destino utilizado por padrão é o 00: e0: 2b: 00: 00: 04. Abaixo vemos o formato do frame do EAPS.


0 1 2 3 4 5
12345678 90123456 78901234 56789012 34567890 12345678
Destination MAC Address (6 bytes)
Source MAC Address (6 bytes)
EtherType
PRI
VLAN ID
Frame Length
DSAP/SSAP
CONTROL
OUI = 0x00E02B
0x00bb
0x99
0x0b
EAPS_LENGTH
EAPS_VER
EAPSTYPE
CTRL_VLAN_ID
0x0000
0x0000
SYSTEM_MAC_ADDR (6 bytes)
HELLO_TIMER
FAIL_TIMER
STATE
0x00
HELLO_SEQ
0x0000
RESERVED (0x000000000000)
RESERVED (0x000000000000)
RESERVED (0x000000000000)
RESERVED (0x000000000000)
RESERVED (0x000000000000)

Com o anel configurado e operando, o Master começa a enviar pacotes de **health check** (pacotes de exame de saúde) pela porta primária a fim de recebê-los pela porta secundária para assegurar que o anel está funcionando. Esses pacotes trafegam pela VLAN de controle do domínio, que é a única não bloqueada na porta secundária. Todas as demais VLANs são bloqueadas a fim de evitar loop no anel.

3.3 Falhas no Anel

O Master pode detectar falhas no anel de duas maneiras, ou pelo não recebimento do pacote de health check (pacotes de saúde) ou pelo recebimento de uma trap de falha enviada por um dos nodos Transit.

Uma vez detectada a falha, o Master desbloqueia a porta secundária para o tráfego de dados das protected VLANs e comunica os Transit que a configuração do anel mudou. Ao receberem o pacote de flush FDB do Master, os Transit começam a aprender a nova configuração da rede. O Master passa a operar no modo Fail.

3.4 Restaurações do Anel

Os pacotes de health check continuam sendo enviados mesmo havendo falha no anel. Quando o anel é restabelecido, o Master recebe novamente o **health check** na porta secundária e então começa a transição para o estado anterior. O tráfego das protected VLANs são bloqueados na porta secundária e os Transit são comunicados de que a configuração do

anel mudou. O Master volta ao modo *complete*. A figura 3.1 a seguir apresenta a máquina de estados do Master.

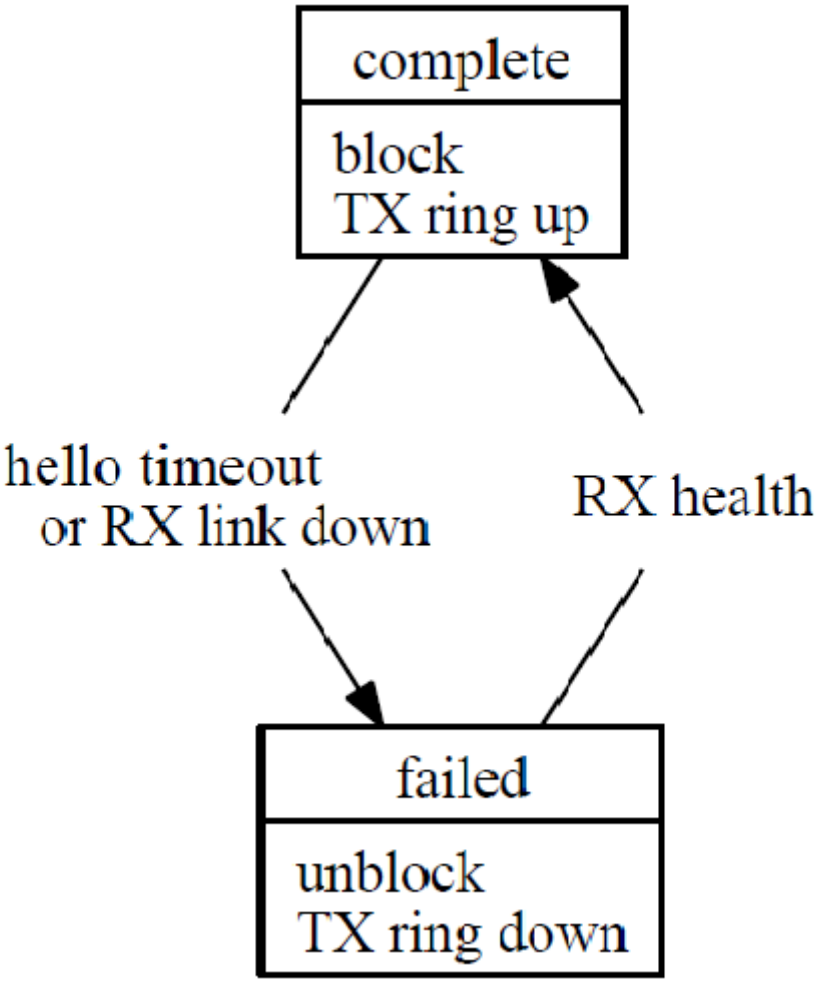


Figura 3.1 Esboço de máquinas de estados do master:

Para o secundário a máquina de estados é um pouco mais complexa.

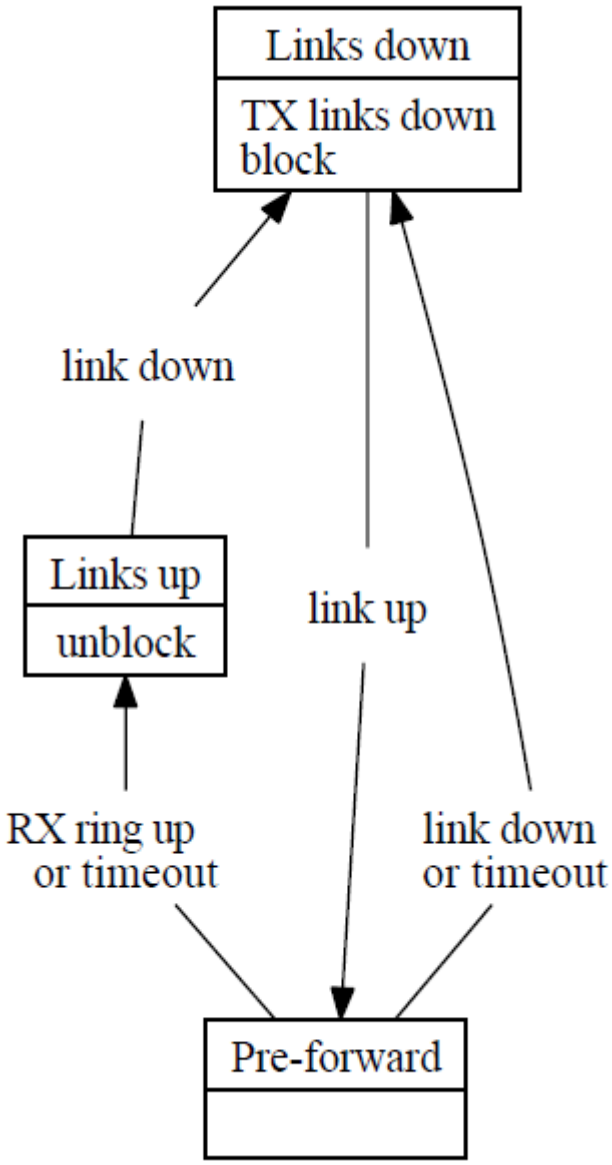


Figura 3.2 Esboço de máquinas de estados do secundário

3.5 Estrutura

Um domínio EAPS é representado pela estrutura st_eaps_domain, que por sua vez engloba outras duas, que são a st_cfg_ eaps_domain e st_status_eaps_domain. A st_cfg_eaps_ domain é a configuração do domínio propriamente dito.

```
typedef struct {
    // domain is created
    unsigned char created;

    // in master mode
    unsigned char is_master;

    // domain name
    char name[33];

    // indexes of member ports in the ring
    int ports[2];

    // period for failure detection. must be greater than hellotime.
    int failtime;

    // period for hello message transmition. must be greater than 0.
    int hellotime;
```

```
        ID. Must be an existent and active VLAN. Must not belong to another domain.
        uint16_t control_vlan;

        // bitmap of protected VLAN groups
        unsigned char protected_vlan_groups[EAPS_MAX_VLAN_GROUPS/8];

} st_cfg_eaps_domain;
```

Já a st_status_eaps_domain é a estrutura que armazena o status do domínio configurado e é utilizada exclusivamente pela máquina de estados do EAPS, não necessita de intervenção da aplicação.

3.6 API

A API da libeaps é bem simples, sendo formada por apenas quatro funções. A eaps_init é a função de inicialização e deve obrigatoriamente ser chamada antes de qualquer outra da API. É através dela que as callbacks e o endereço MAC são passados à libeaps.

A eaps_loop é a função principal da libeaps. É um loop infinito que executa a cada segundo a máquina de estados do master ou do transit de cada domínio EAPS configurado. A função obtém esses domínios através de chamadas a callback eaps_get_ domain definida na aplicação. A eaps_loop deve ser executada num processo ou thread específica, pois nunca retorna.

A eaps_ rx deve ser executada pela aplicação a cada vez que for recebido um pacote EAPS e a eaps_port_link_event a cada vez que mudar o status de um link.

3.7 Callbacks EAPS

As callbacks são divididas em dois grupos, as de eaps e as de debug. As callbacks do eaps são definidas no typedef_st _eaps callbacks em eaps.h e são todas obrigatórias.

Algumas funções fazem referência à ”index” do domínio. Esse index na verdade é um número único que identifica exclusivamente um domínio e, dependendo da aplicação, pode ser a posição do domínio numa lista de domínios, daí o uso da palavra ”index”.

3.8 Múltiplos Domínios EAPS por Anel

Cada domínio EAPS tem seu próprio nó mestre e a sua própria VLAN de controle, e ainda seu próprio grupo de VLANs de área protegida. Diferentes domínios EAPS poderão ter seus mestres no mesmo switch ou em chaves diferentes. Além disso, vários domínios EAPS podem coexistir no mesmo anel. Este recurso permite o EAPS tirar proveito dos recursos disponíveis e largura de banda no anel, chamado de reuso espacial. Isso proporciona a flexibilidade para controlar cada grupo de VLANs independentemente, portanto, tornando a banda mais eficientemente. Além disso, um domínio pode conter VLANs com os clientes nas proximidades permitindo caminhos mais diretos entre nós e controlar a direção do fluxo de tráfego. Veja na figura 3.3 abaixo o uso de mútiplas VLANs por anel, nessa figura estão demonstrados dois domínios EAPS, sendo testados por um gerador de tráfego os dois sentidos do anel.

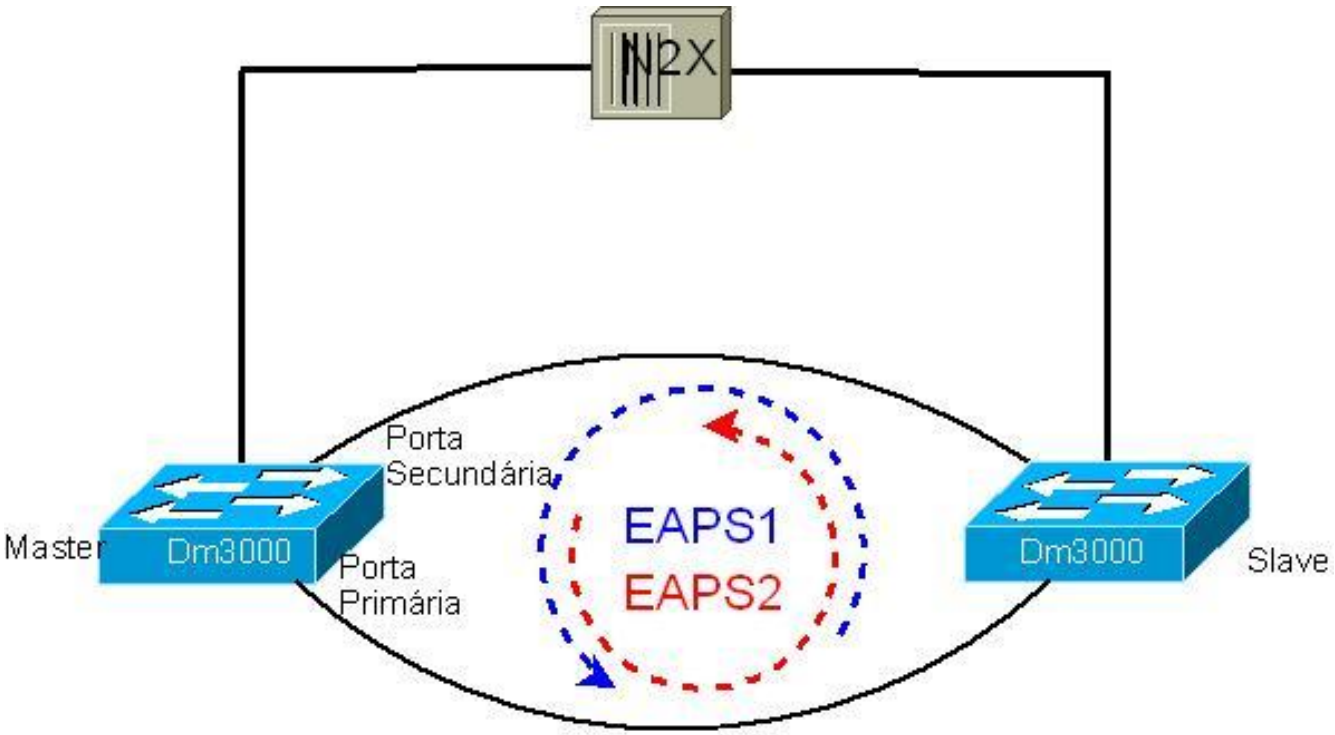


Figura 3.3 múltiplas instâncias

3.9 Múltiplos Anéis

Um switch pode servir para interligar múltiplos anéis. Essa facilidade faz com que haja interoperabilidade entre diversos anéis e domínios distintos. Veja na figura 3.4 as múltiplas interconexões de rede possíveis, podendo ser configurado ainda múltiplas instâncias em cada anel, com o intuito de melhorar ainda mais a performance. As torres em preto na figura 3.4 representam, por exemplo, as ERBs (Estações Rádio Base) que são os elementos de circuito que são protegidos pela resiliência EAPS.

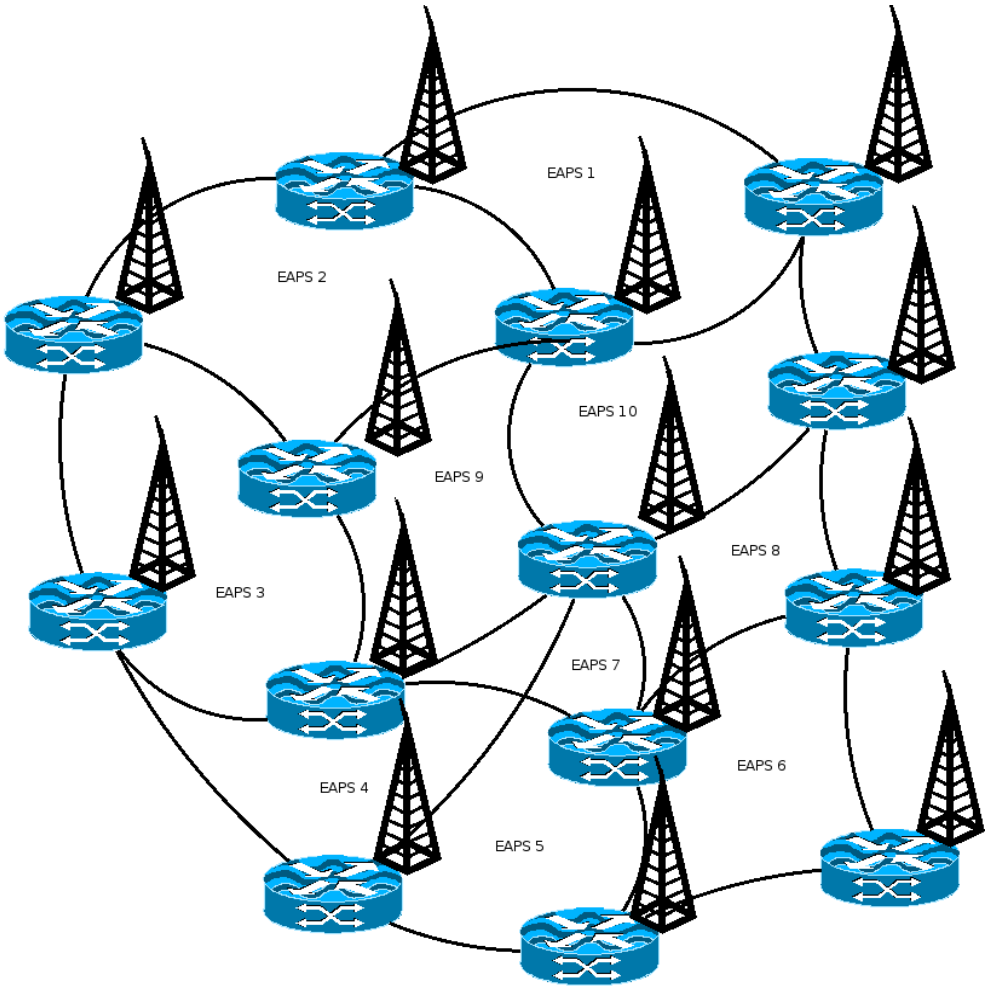


Figura 3.4 múltiplos Anéis

3.10 Múltiplos elementos por anel

O EAPS permite que num mesmo anel seja inserido até 20 elementos sem que haja perda de performance significatica. O cenário abaixo foi testado através do appliance N2X da Agilent, com um aplicativo verificador de performance. As tabelas de performance serão apresentadas a seguir.

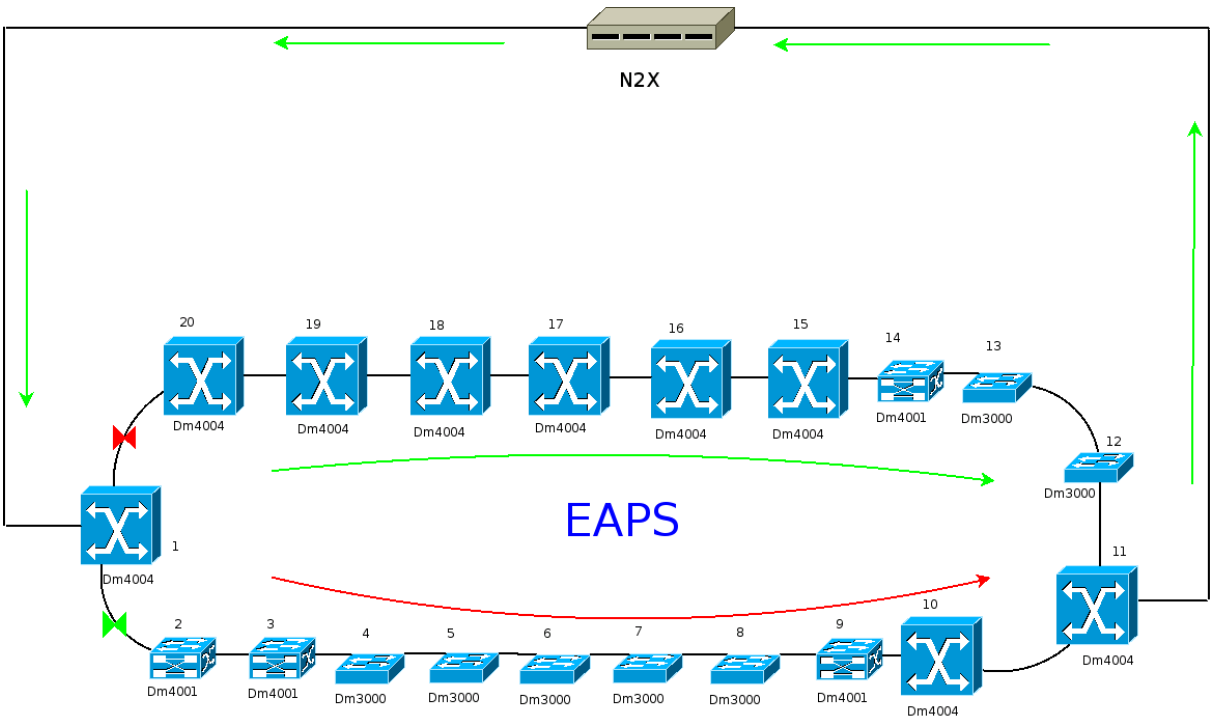


Figura 3.5 Anel com 20 elementos

3.11 Configuração do EAPS

Para a configuração de um domínio EAPS é necessário definir os seguintes parâmetros:

1) Declarar o domínio EAPS:

```
DmSwitch3000(config)#eaps 1
```

2) Definir o elemento que será o master e os demais transits da topologia:

```
DmSwitch3000(config)#eaps 1 mode master/transit
```

3) É recomendado que seja colocado um nome para o domínio:

```
DmSwitch3000(config)#eaps 1 name XXXXXX
```

4) Definir as portas pertencentes ao anel, lembrando que o master será quem definirá o sentido do tráfego.

Os elementos transit não definem o sentindo do tráfego, dessa forma não há influência de qual porta será selecionada como primária ou secundária nesses elementos.

```
DmSwitch3000(config)#eaps 1 port primary ethernet 1/28
DmSwitch3000(config)#eaps 1 port secondary ethernet 1/27
```

Criar a VLAN de controle:

```
DmSwitch3000(config)#eaps 1 control-vlan id 4094
```

As portas primária e secundária devem ser membros da VLAN de controle.

```
Dm_104.5(config)#interface vlan 4094
Dm_104.5(config-if-vlan-4094)#set-member tagged ethernet range 27 28
```

Definir qual grupo de VLANs devem ser protegidas:

```
DmSwitch3000(config)#vlan-group 2
DmSwitch3000(config)#vlan-group 2 vlan range 2 4000
```

Proteger o grupo de VLANs desejado pelo EAPS:

```
eaps 1 protected-vlans vlan-group 2
```

No Dmswitch3000 é possível adicionar até 16 domínios EAPS e no Dm4000 até 64 domínios.

A verificação de todos parâmetros configurados, para o EAPS, pode ser feita com o comando `show eaps detail`. A resposta ao comando é demonstrada a seguir.

```
DmSwitch3000(config)#show eaps detail
Domain ID:                    1
Domain Name:                  TESTE
```

```
State:                          COMPLETE
Mode:                           Master
Hello Timer interval:           1.0 sec
Fail Timer interval:            3.0 sec
Pre-forwarding Timer:           6 sec (learned)   Remaining:   0 sec
Last update from:               00:04:DF:11:11:11, Eth 1/27, Sat May 3
16:30:38
Last seq. number received:      14291
Primary port:                   Eth1/28           Port status: Up
Secondary port:                 Eth1/27           Port status: Up
Control VLAN ID:                4094
Protected VLAN group IDs:       2
```

4 MELHORIAS REALIZADAS NO EAPS

Num primeiro momento foi necessário identificar todas as características do protocolo EAPS, com a finalidade de aprimorar suas deficiências e atingir a meta de ter o protocolo com tempos de comutações inferiores a 50ms.

Foi utilizado como ferramenta de medição o appliance N2X da Agilent Technologies, exibido na figura 3.6. O teste seguiu a topologia e pré-condições recomendadas pelo fabricante do equipamento de teste, que solicita uma porta do N2X ligada diretamente ao master, enviando um tráfego para outra porta do N2X, que estará ligada em outro switch da topologia, como mostrado na figura 4.1. O teste inicialmente irá utilizar um profile com 1.000 fps de 64 bytes, posteriormente 1.000.000 fps de 64 bytes cada.

Figura 4.1 N2X da Agilent Technologies

Abaixo, na figura 4.2, vemos a topologia recomendada pelo fabricante do N2X para que seja garantido o desempenho do equipamento nas medidas.

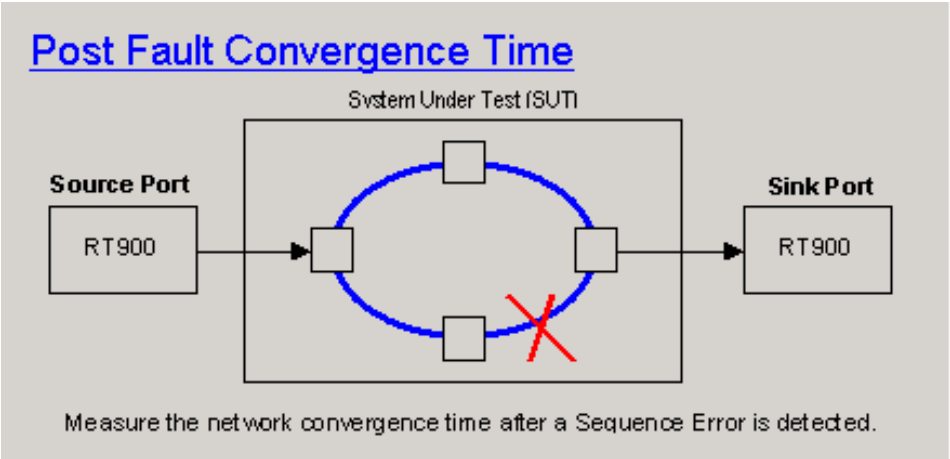


Figura 4.2 Recomendação de topologia para teste de desempenho com o N2X

Os resultados foram comparados ainda com o appliance da Spirent o “Smart Bits” exibido na figura 4.3. Neste equipamento não há recomendações de topologia nem um aplicativo específico para medir tempo de convergência. Como o desempenho foi praticamente idêntico entre os testadores, foi preferido o teste com o N2X, por possuir uma ferramenta específica e ainda gerar relatório de medidas.

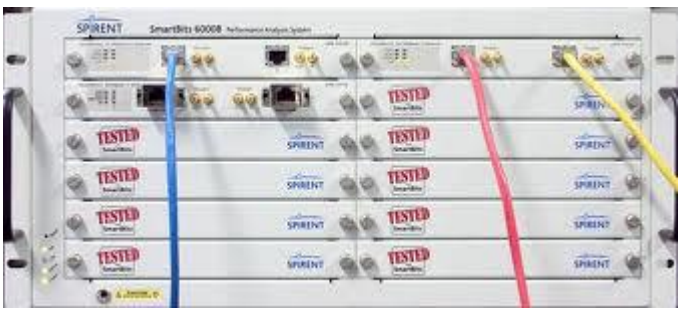


Figura 4.3 Smart Bits da Spirent.

Na figura 4.4 é mostrada a topologia que foi submetida ao teste de desempenho. O relatório de medidas é demonstrado a seguir pela fig 4.5, relembrando que essas medidas foram realizadas com o firmware sem as melhorias de performance de convergência. O teste foi feito com 2 DmSwitch3000. No teste foi gerado 1 MAC de origem X 1 MAC de destino, o tráfego gerado foi de 1 gigabits/segundo com pacotes de 64bytes. O motivo do uso de pacotes de 64bytes é para justamente testar os casos de dados mais críticos como, por exemplo, os de Voip. O Overhead nesse caso é maior, devido ao tamanho reduzido dos pacotes [13]. Os pacotes saíram com prioridade 7, ou seja, pacotes com alta priorização, cujo objetivo é validar o comportamento do DmSwitch nesta situação.

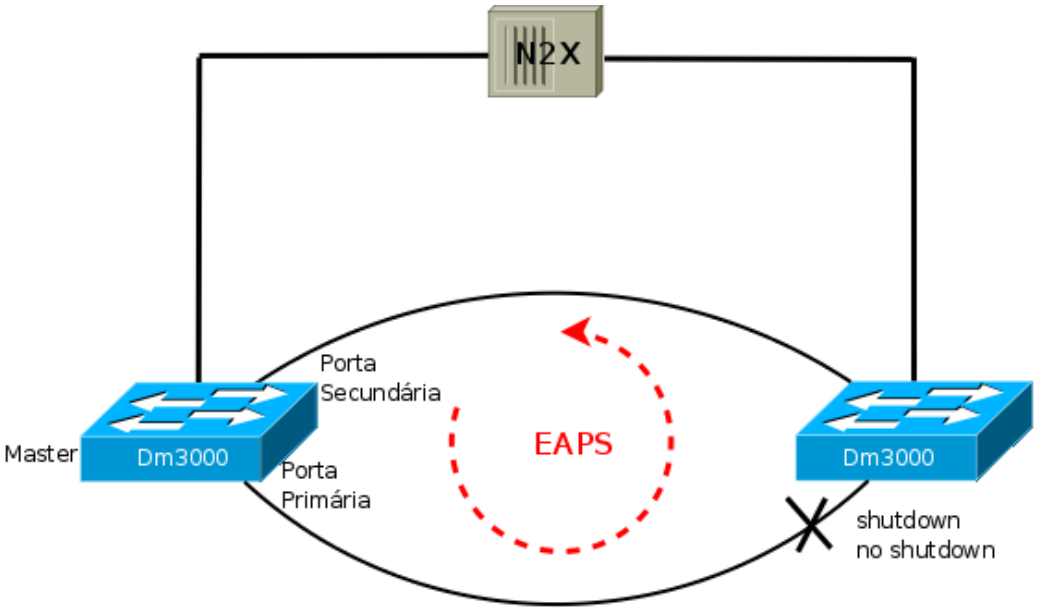


Figura 4.4 topologia do teste de performance

==========

Agilent Technologies N2X - Test Report

Copyright 2009 Agilent Technologies

==========

Script : PostFaultConvergenceTime v1.3

Tester software : RouterTester900 6.11 System Release

QuickTest software : 7.10 (12-Aug-2009)

Run started : Mon Nov 01 08:57:04

Test engineer :Adriano Reis

System under test : Router: X, Software: Y

==========

Test Parameters

==========

Monitored port : 101/3, TRI_RATE_ETHERNET_X

Minimum expected packet loss : 5

Test duration : Run continuously

Stop after first fault : no Results

=======

Port Name Convergence Time (ms) Lost Packets

--- --- ---

101/3 70.816050 30981

101/3 73.085590 33224

101/3 70.985110 48052

101/3 70.000920 4987

101/3 70.669550 39667

101/3 79.624890 26085

101/3 78.557650 69118

101/3 -3.000920 -1 (below minimum expected)

101/3 -10.000900 -1 (below minimum expected)

101/3 73.571100 38757

101/3 73.858500 36687

101/3 74.482200 6335

101/3 77.874700 25319

101/3 77.693600 7740

101/3 74.912000 8398

101/3 73.565500 49684

101/3 78.175900 32451

101/3 75.354200 779

101/3 70.440200 4379

101/3 72.406700 9802

101/3 73.565500 49684

101/3 78.175900 32451

101/3 73.858500 36687

101/3 74.482200 6335

101/3 77.874700 25319

==========

Run Ended: Mon Nov 01 09:10:08 2010 (duration = 22:03:00)

==========

Este teste preliminar demonstrou um tempo de convergência superior a 70ms. Indicando que o tempo de convergência está acima do tempo desejado . Nesta mesma tabela notamos o incremento de pacotes, ou seja, número de pacotes negativos. Isso é uma indicação de ocorrência de loops durante os chaveamentos.

Fazendo um debug do protocolo com um firmware que imprime a troca de pacotes no CLI ,é visto que há perda de pacotes de controle do EAPS. Os pacotes de saúde são perdidos e o estado do EAPS fica alternando de COMPLETE para FAIL e de FAIL para COMPLETE. Abaixo parte do debug realizado.

11/09/2010 19:47:32.81 <Info:EAPS.DmnInfo> EAPS - Fail-timer-exp flag cleared.

Domain state: Complete

11/09/2010 19:47:32.81 <Info:EAPS.DmnInfo> EAPS - Fail-timer-exp flag set.

Domain state: Fail. Health check loss

11/09/2010 19:47:28.86 <Info:EAPS.DmnInfo> EAPS - Fail-timer-exp flag cleared.

Domain state: Fai. Health check loss

11/09/2010 19:47:24.82 <Info:EAPS.DmnInfo> EAPS - Fail-timer-exp flag set.

Domain state: Complete

11/09/2010 19:47:17.81 <Info:EAPS.DmnInfo> EAPS - Fail-timer-exp flag cleared.

Domain state: Fail. Health check loss

11/09/2010 19:47:17.81 <InfO:EAPS.DmnInfo> EAPSD DSK33 - Fail-timer-exp flag set.

Domain state: Complete

Alterando a prioridade do tráfego nas interfaces geradoras do N2X de 7 para 6, as perdas de pacotes de controle/saúde do EAPS pararam de ocorrer. Logo foi presumido que os pacotes de controle estão competindo com os pacotes de dados. Isso não será crítico caso o usuário compreenda que isso é uma característica do equipamento. Pois serão raros os casos que o usuário tenha 100% de utilização da porta do DmSwitch com tráfego de alta prioridade, mas isso poderia comprometer a confiabilidade no protocolo, caso o usuário não estivesse atento para tal fator. Foi preferido então tratar também essa característica de maneira que os pacotes de saúde entrem em uma fila de "strict priority". Isso foi feito no próprio BCM do CI de switching. Quando o usuário configurar o EAPS identificando a VLAN pertencente ao domínio de controle , estas VLANs serão tratadas como SP – Strict priority-, ou seja, os pacotes de controle de EAPS terão garantia mínima de banda. Isso garante que mesmo que o usuário faça essa configuração para seus pacotes, o CI de Switching irá garantir essa reserva de banda exclusivamente para pacotes originados por essa porta.

Cada frame de controle tem 2bytes , logo tornou-se fácil a reserva de banda para esse tráfego, considerando ainda que somente o switch master gera esses pacotes e os switches transits encaminham os mesmos pacotes de volta ao master. Como o CLI irá permitir o envio de pacotes em múltiplos de100milisegundos, foi previsto para pior caso, que é o envio de 1 pacote a cada 100ms. Logo 16bits X 10 daria um total de 160bits/segundo, no máximo. Mas como a granularidade do BCM é de 64bits/segundo, foi escolhido a reserva de 196bits/segundo. Por exemplo: quando o usuário configurar a porta 26 como pertencente ao

domínio EAPS a mesma irá ficar com reserva de banda de 196bits/segundo, não sendo exibido pelo CLI esta reserva. Essa reserva de banda torna-se insignificante frente a capacidade da porta que se alterna entre 1gigabits/segundo até 10gigabits/segundo dependendo do hardware adquirido pelo cliente. As linhas de comando no BCM (shell do CI) de switching, são exclusivas desse fabricante.

Num segundo momento foi iniciado um debug na topologia para minimizar o tempo de comutação. Foi implementado um comando para permitir que o usuário do CLI configure o tempo de envio de pacotes de saúde. *Hello Timer interval* -que é o período dos pacotes transmitidos, que verificam o status do EAPS e o *Fail Timer interval* que é o tempo máximo que irá aguardar a chegada de um pacote de hello sem que seja alterado o status do EAPS.

Foi alterado a função *eaps_loop* que é a função principal da LibEAPS cujo propósito é um loop infinito rodando a cada segundo. A implementação foi tratada também nas sub rotinas, pois afeta o tempo de *Hello Time* e *Fail Time*. A nova rotina seguiu a estrutura da já existente, para respeitar que o intervalo de hello time fosse inferior ao de fail time, a fim de que o usuário não gere uma situação em que os switches fiquem em modo de falha por um erro na configuração do CLI. Uma mensagem será exibida ao usuário caso a condição *Hello Time* < *Fail Time* não seja satisfeita, informando que os tempos continuarão no modo padrão: fail em 3 segundos e hello em 1 segundo.

A partir de agora será possível configurar o *hello time* e *fail time* em frações de segundo, com intervalos de 100milisegundos, como mostrado abaixo:

```
DmSwitch3000(config)#eaps 1 hellotime 0 milliseconds
0-900  Fraction of seconds (valid only for seconds equal to zero)

DmSwitch3000#eaps 1 failtime 0 milliseconds
0-900  Fraction of seconds (valid only for seconds equal to zero)
```

Após comitar essa configuração e atualizar os firmwares dos switches envolvidos, foi verificado que houve um pequeno aumento de processamento quando aumentado o número de pacotes de saúde (diminuindo o hellotime para 100ms) o aumento no consumo de CPU pelo processo RX_PKT foi de aproximadamente 3%, pois diminuir o intervalo de pacotes de hello fez obviamente com que a CPU tivesse que tratar mais pacotes. Esse aumento de consumo de CPU não chega a ser considerado de muita expressão, mas vai de encontro ao que é buscado pelo equipe que desenvolve protocolos L3, que é liberação de recursos da caixa, tanto de memória como de CPU para a implementação de novos protocolos. A alternativa de aumentar o número de pacotes de hello continuava sendo uma boa escolha para aprimorar a velocidade de convergência, já que houve um ganho no desempenho de comutação. O ganho ainda estava aquém do desejado após o teste, mas foi percebido uma melhora de 10ms em média, em comparação ao teste anterior. Essa solução não poderia ser descartada. Ocorreu daí a idéia de tentar fazer com que os pacotes de hello não fossem respondidos pela CPU e sim pelo CI de switching. Novamente uma alteração no código: Como o MAC de controle do EAPS é conhecido e sempre o mesmo, os pacotes enviados para esse endereço poderiam ser respondidos como é feito para outros pacotes unicast. Em tráfegos L2 todos os pacotes com origem e destinos já conhecidos são tratados pelo CI de switching . Então pareceu ótimo seguir esse mesmo padrão para o EAPS. Algumas alterações em funções como: *packets_unknow* e *table_mac_initd* , em relação ao MAC `00: e0: 2b: 00: 00: 04,` fez com que os frames enviados para esse MAC fossem respondidos diretamente pelo CI de switching. Essa alteração não iria impactar na melhoria da performance, mas sim garantir que em caso de indisponibilidade da CPU o protocolo ainda seria confiável.

Dar o mesmo tratamento dos pacotes unicasts para frames de controle chamou a atenção para um novo estudo. No momento da convergência, seja por perda de link ou por

shutdown em uma interface é realizado flush da tabela L2. Isso faz com que haja uma rajada de pacotes broadcast até que a tabela MAC (*mac-address-table*) seja preenchida novamente. Esta rajada é controlada por default em 500pps (500 pacotes por segundo). Logo esse controle de 500 pps pode estar fazendo com que haja uma demora no aprendizado dos MACs, por consequência atrasando a convergência do EAPS. Um teste inicial pôde ser feito diretamente através do CLI, com o comando *no switch-port-storm-control broadcast.*

Após análise do comportamento do storm-control a implementação foi feita no firmware de uma maneira mais inteligente, pois não é seguro deixar que seja feito flood de pacotes para a CPU. Então a implementação foi feita removendo o storm-control apenas no momento do flush da tabela, evitando que a CPU seja atacada pela inundação de broadcast.

5 RESULTADOS ALCANÇADOS

Após comitar as alterações realizadas numa nova imagem de pré-firmware, foram atualizados ambos os DmSwitches, representados na fig 4.4. O pré-firmware apresentou um ótimo resultado, conforme visto no relatório abaixo. A priorização dos pacotes com garantia de banda, trouxe ainda mais garantias para o cliente.

Quanto a questão de “flaps” (alternar de modo complete para fail e vice versa) . O teste com o N2X, confirmou a confiabilidade da nova versão de firmware, no que diz respeito a “flaps”. O tratamento dos frames de saúde pelo CI de switching ao invés da CPU garantiu maior robustez ao protocolo, garantindo o funcionamento do protocolo mesmo em casos de indisponibilidade da CPU.

Ao tornar possível a configuração dos pacotes de saúde, trouxe apenas maior flexibilidade ao protocolo. O impacto dessa alteração foi praticamente imperceptível no ponto de vista de tempo de comutação, porém o envio de Traps ou alarme de gerência serão mais instantâneos quando houver queda de um circuito.

A alteração mais impactante foi a de liberar o storm-control durante a comutação. Houve uma maior velocidade na capacidade de preenchimento da tabela MAC, fazendo com que a comutação caísse para a casa dos 35ms, em média. Isso realmente foi o que rendeu a maior performance.

Abaixo são apresentados os resultados do teste realizado na mesma condição supracitada.

```
================================================================
Agilent Technologies N2X - Test Report
Copyright 2009 Agilent Technologies
================================================================
Script : PostFaultConvergenceTime v1.3
Tester software : RouterTester900 6.11 System Release
QuickTest software : 7.10 (12-Aug-2009)
Run started : Mon Nov 14 12:55:08 2010
Test engineer :Adriano Reis
System under test : Router: X, Software: Y
================================================================
Test Parameters
===============
Monitored port : 101/3, TRI_RATE_ETHERNET_X
Minimum expected packet loss : 5
Test duration : Run continuously
Stop after first fault : no
Results
=======
Port Name Convergence Time (ms) Lost Packets
--------- -------------------- ------------
101/3     34.517000   41642
101/3     33.505500   39684
101/3     30.175900   32451
101/3     33.753100   38779
101/3     30.123400   33734
101/3     35.956700   43785
101/3     36.765500   45687
101/3     35.275900   43435
101/3     29.858500   31562
101/3     31.541200   34734
101/3     29.548500   31647
101/3     32.525900   36335
101/3     30.774700   32531
101/3     38.571100   33875
101/3     29.858500   31287
101/3     27.340200   30361
101/3     31.306500   29802
101/3     31.554300   34491
101/3     30.452700   33451
101/3     30.187700   32533
101/3     31.653700   34212
101/3     30.571100   32534
101/3     37.968500   33497
101/3     30.175900   32456

================================================================
Run Ended: Mon Nov 15 13:22:19 2010 (duration = 24:12:32)
================================================================
```

6 CONCLUSÕES

Este trabalho proporcionou aprimorar os conhecimentos adquiridos durante o curso. O resultado buscado foi alcançado, conseguindo tempos de comutação menores que 50ms e ainda contar com um protocolo de resiliência seguro.

O projeto ainda deixa possibilidades de melhorias para serem estudadas. Desde o protocolo até métodos de aperfeiçoamento de medidas de performance em laboratório.Por exemplo, foi visto que num teste de performance de convergência não é recomendado o uso de cabo elétrico, pois a autonegociação é obrigatória neste meio físico, e a troca de páginas da autonegociação irá comprometer a performance do link. A autonegociação ainda irá permitir o envio de pause-frames caso o flow-control esteja habilitado, prejudicando a performance do teste, logo recomenda-se o uso de cabos óticos com velocidade forçada em 1000full. O número de VLANs criadas para o teste também pode ser alvo de estudo mais aprofundado. A análise com a geração de MACs aleatórios. O estudo do tamanho da tabela de memória pode aprimorar ainda mais os tempos encontrados. A análise de comportamento em novos hardwares e em placas com possibilidade de stacking são novos desafios, pois o processamento para essas unidades é centralizado em uma única placa controladora.

BIBLIOGRAFIA:

[1] Abdul Jabbar Mohammd, David Hutchison, and James PG Sterbenz, 14th IEEE international conference on Network Protocols (ICNP 2006), Santa Barbara, Califórnia, USA November 2006.

[2] IETF RFC – 3619 (EAPS),

[3] IEEE 802.1D (STP), 1990.

[4] IEEE 802.1w (RSTP), 1998.

[5] IEEE 802.1 S (MSTP), 2006.

[6] ODOM, W**. CCNA Self-Study CCNA INTRO Exam Certification Guide**. Indianápolis, Cisco Press, 2004, 627 p.

[6] BOYLES, T. HUCABY, D. **Cisco CCNP Switching Exam Certification Guide** **I**ndianapolis: Cisco Press, 2001.576 p.

[7] IEEE 802.1Q (VLANs), 1990.

[8] IEEE 802.1s (MSTP), 2003.

[9] CLARK, K; HAMILTON, K. **Cisco LAN Switching.** 1 ed. Cisco Press, 1999. 960 p.

DIOGENES, Y. **Certificado cisco: CCNA 4.0:** guia de certificação para o exame 640-801. 3 ed. Rio de Janeiro: Axcel Books, 2004, 408 p.

[10] HUCABY, D. **CCNP Self-Study CCNP BCMSN Exam Certification Guide.** 3 ed.Indianapolis: Cisco Press, 2006. 618 p.

[11] LAMMLE, T. **CCNA: Cisco Certified Network Associate:** Study Guide. 6 ed. Indianapolis, Sybex 2007. 1012 p.

[12] ODOM, W. **Guia de certificação do exame Cisco CCNA.** Tradução de Eduardo Messia Oliveira. 3 ed. Indianápolis: Cisco Press; Rio de Janeiro: Alta Books, 2003, 738 p.

[13] PERLMAN, R. **Interconnections Bridges, Routers, Switches, and Internetworking Protocols**. 2 nd. 418 p.

[14]CCITT Recommendation, **Definitions relating to echo suppressors and characteristics of a far-end operated, differential, half-echo suppressor, Blue Book,** Vol. III, Rec. G.161, ITU, Geneva, 1965.

[15] COMER, Douglas. **Computer Networks and Internets** – 2 ed. Prentice Hall Inc, 1999.

[16] DIÓGENES, Yuri. Certificação Cisco – **CCNA 3.0 Guia de Certificação para o Exame #640-607** – 2002, 2ª Edição, Axcel Books do Brasil Editora Ltda, 2002.

[17] FEIBEL, Werner. **Encyclopedia of Networking**. SYBEX Inc, 1996.

[18] FURUKAWA, **FURUKAWA. Data Cabling System**. Guia Didático. Curso de Cabeamento Estruturado. Curitiba, 2003.

[18] JACK Terry. **CCNP: Building Cisco Multilayer Switched Networks**. SYBEX Inc, 2003.

[19] KUROSE & ROSS, James F.; Keith W. ROSS. **Rede de computadores e a Internet**: uma nova abordagem; Tradução Arlete Simille Marques; revisão técnica Wagner Luiz Zucchi – 1ª Edição – São Paulo : Addison Wesley, 2003.

[20] HALABI, Sam , **Cisco Internet Routing Architectures 2 ed.** Cisco press metro ethernet - the definitive guide and carrier metro ethernet applications